

Game Programming I

[Term Project]

June 16, 2005

19984143 서무교

19984153 안상근

Hallym University

목 차

- 게임 설명
- 구현 환경 및 방법
- 특징 / 협업 작업내용
- Screen Shot / Demo 동영상

Hallym University

2

게임 설명

- 게임제목 : 3D 알까기
- 게임 시나리오
 - 공격하는 돌이 충돌하는 돌들을 장기판으로부터 떨어트리는데 충돌하는 돌들을 모두 떨어트리면 게임이 종료된다.
 - 공격하는 돌이 충돌하는 돌들을 장기판으로부터 다 떨어트리지 못하고, 장기판으로부터 떨어지면 게임이 종료된다

게임 설명

- 게임 방법
 - 상, 하, 좌, 우(방향키)를 조절하여 공격하는 돌의 방향을 정해진 뒤 **SpaceBar** 키를 눌러 해당 방향으로 움직이게 하여 돌들을 장기판으로부터 떨어뜨린다.

- 구현 환경
 - Visual C++ 6.0, DirectX 9.0 2003 Summer Version
- 구현 방법
 - 수업시간에 다루었던 Sample code들을 기초로 구현
 - 사용한 부분
 - Light (directional, spot), Texture mapping, Font, Material, CreateCube (indexbuffer, vertexbuffer), CreateSylinder, Filtering, etc...

- 프로그램 구성
 - DrawFont()
 - 화면에 폰트를 출력하는 부분
 - DrawItem()
 - 게임에 필요한 요소들 setting, Render하는 부분
 - Process()
 - 사용자로부터의 키 값을 처리하는 부분
 - MoveStone()
 - 충돌체크 부분
 - Display()
 - Keyboard input 값으로 update하는 부분

구현 환경 및 방법

```
//돌이 충돌체크를 하면서 이동 한다.
void MoveStone(int ActStone, D3DXVECTOR3 Dir, float dPower)
{
    int cnt = 0;

    if(dPower<=0)
        return;

    D3DXVECTOR3 vSourcePos, vTargetPos;
    D3DXVECTOR3 vColDir;

    vSourcePos.x = WorldMatrices[ActStone]._41;
    vSourcePos.y = WorldMatrices[ActStone]._42;
    vSourcePos.z = WorldMatrices[ActStone]._43;

    while(dPower>0) {
        // 공격하는 돌
        WorldMatrices[ActStone]._41 += Dir.x * dPower;
        WorldMatrices[ActStone]._42 += Dir.y * dPower;
        WorldMatrices[ActStone]._43 += Dir.z * dPower;

        //충돌 체크
        for(int i=0; i<4; i++) //충돌 체크할 돌이 0-3번돌
        {
            if( i == ActStone ) //공격돌과는 충돌 체크할 필요가 없음
                continue;

            //충돌 체크할 돌의 Pos를 구한다
            vTargetPos.x = WorldMatrices[i]._41;
            vTargetPos.y = WorldMatrices[i]._42;
            vTargetPos.z = WorldMatrices[i]._43;
        }
    }
}
```

위치를 계산할 Vector 선언

공격하는 돌의 x, y, z좌표

이동한 후 공격하는 돌의 x, y, z좌표

공격받는 돌들의 x, y, z좌표

구현 환경 및 방법

```
//공격돌의 Pos와 충돌 체크할 돌의 Pos 의 거리를 계산
vColDir = vTargetPos - vSourcePos;
if( D3DXVec3Length(&vColDir) <= 4)
{
    //충돌났어!!! 각종 물리적인 현상이 있겠지비~
    //Pos이 판의 범위를 벗어나면... 해당 돌을 보내자..어디로?..
    if( vTargetPos.x > 20 || vTargetPos.x < -20 ||
        vTargetPos.z > 20 || vTargetPos.z < -20 )
    {
        D3DXMatrixTranslation(&WorldMatrices[i], 20, 0, -25);
        D3DXMatrixScaling(&Pscale, 0.7f, 0.7f, 0.7f); // 장기알 scale
        D3DXMatrixRotationX(&rPos, 3.14f * 0.05); // 장기알 Rotation
        WorldMatrices[i] = Pscale * rPos * WorldMatrices[i];
        cnt++;

        if( cnt == 3 )
            end = true;
    }
}
```

충돌체크 부분

특징 / 협업 작업내용

- 특징
 - Simple한 Interface
 - 간편한 게임 조작
- 협업 작업내용
 - 안상근
 - 장기알, 장기판, 배경 생성 및 Texture mapping
 - 서무교
 - 장기알 충돌체크, 함수 모듈화

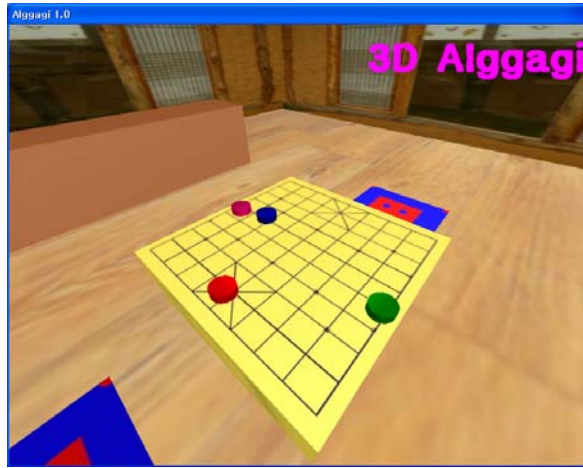
Screen Shot / Demo 동영상

- 시작화면



Screen Shot / Demo 동영상

■ 게임화면



Screen Shot / Demo 동영상

■ 종료화면

